The Wireless-Mobility Integration Protocol (WMIP) For Using TCP/IP Over Wired And Wireless Networks

Robert Signorile
Kevin Lester
Boston College
Computer Science Department
Fulton Hall 460
Chestnut Hill, MA 02467
Email: signoril@bc.edu
Phone: 617-552-3936

KEYWORDS
Computer Networks, Hybrid Simulation, Telecommunications

ABSTRACT

This paper describes a protocol, which helps standardize and plug some of the holes in today's wireless network protocols. All of the standard protocols available today all use some form of data modulation to physically send packets over the airwaves in such a way as to have the packets redundant and be able to sustain some interference. But even using the best modulation algorithm, packets are always susceptible to environmental noise and interference. According to (Conover 2000) 802.11b, one of the wireless network standards available today is only 85% efficient on its physical layer. TCP works relatively well at low packet loss levels, and operates very poorly with larger packet losses. At noisy networks, TCP backs off its transmission windows to operate basically like a stop-and-wait algorithm. Today's Internet has about a 5% packet loss, as can be seen at (The Internet Weather Report) so TCP works relatively smoothly. But at 15% packet loss, wireless networks experience poor throughput due to TCP's congestion control mechanisms. None of today's wireless protocols handle handoffs either, (except the cellular networks because it is inherent in their systems), nor do they specify how to handle handoff across routers, which is the big problem. This paper introduces the WMIP protocol, which inherently boosts TCP performance over wireless networks, specifies how to handle handoffs between APs, and also handles handoffs between routers. Currently, this is left to the hardware manufacturers, which could lead to inoperability between hardware manufacturers. In this paper we present a modification protocol for TCP/IP over wired and wireless networks. We call this the Wireless-Mobility Integration Protocol (WMIP). We have simulated this protocol and have show it to be better than standard TCP/IP over wireless networks, with the added benefit that we have unified TCP for both wired and wireless environments

INTRODUCTION

The current version of IP (version 4) assumes that each node is fixed and that all packet losses are due to congestion. This being the case, TCP backs off its transmission window when a loss occurs, and does a *slow-start.* In the wireless world, this is not the case. All packet losses are not necessarily due to congestion, but simply due to environmental interference. All wireless network protocols implement some sort of packet modulation (such as Orthogonal Frequency Division Multiplexing (OFDM)), which helps to curb as much environmental interference as possible, but at best, the most efficient wireless network is still only 85% efficient. This means that there will be the need for many retransmissions, in essence generating decreased throughput.

However, if a buffer is put on the AP (Access Point), then the retransmissions can be handled locally, thus saving the FH (Fixed Host) the trouble of resending from its location. This would not only help the throughput of TCP with a mobile host, but would also help curb packet loss when a handoff occurs. The AP could simply buffer each incoming packet and forward them onto the mobile host. If an ack is not received by some timeout period, then the AP could simply resend the data locally. If a mobile host is in a transmission period, then the new AP could send the old AP a packet asking it to forward its data onto the current AP. This would greatly curb the packet losses due to handoffs.

This, however, is impractical because if a handoff takes a long time to complete, then, even though there are no packet losses, the TCP sender could timeout several times before the MH (Mobile Host) successfully hands off to a new AP. If handoffs occur frequently, packet throughput would be very low. There is also a question of the amount of packet buffering, and how big it should be. With long handoffs, there could be a huge amount of buffering that would eventually lead to losses of packets, or to a crashing of the AP. This, of course, is unacceptable.

To solve this problem would require some sort of TCP aware solution, whereas TCP is notified of the cause of the packet loss, so it doesn't resort to a stop and wait algorithm. There have been a great deal of recent research in this area, including (Gerla et. al.), (Balakrishnan et. al.),(Fitzek et. al),(Chan et. al.).
But the purpose of this paper is to make a TCP unaware solution to this problem, so that the local problems are handled locally. Luckily, TCP already has an allowance for exactly what we want to occur. In normal operation, an ack is sent back to the sender with the current window size in the packet. If a packet is sent with a window size of 1, however, TCP will freeze its sending window and enters "persist" mode. This means that TCP receives a message saying that the client's window size is 0, so it freezes its timeout timer, and stops all activity. It simply waits and sends out a probe every so often to make sure that the client is still alive. TCP will then resume its normal operation when it receives an ack with the original advertised window value, or a new ack is received with the latest advertised window value.

This works out well because it allows the FH to stop its transmission during handoffs and very noisy times which will negate the over buffering problem. It also allows the FH to continue sending when conditions are more favorable with the same window size, so throughput is not compromised.

Locally, a protocol that operates in the way described above will solve many of the problems facing TCP over wireless. However, there is still a huge problem of inter-router handoffs, which occurs when mobile hosts wander out of their "home" subnets. Right now, this problem is handled by the implementation of mobile TCP. Mobile TCP specifies that mobile nodes specify a "home agent" which is originally assigned an IP address, and this is the IP address that all incoming packets for that mobile node are sent to. Should the mobile node leave its home address, then an "away agent" is created which then registers with the home agent, and the home agent forwards on all packets it receives to the new address. This, however, is not a standardized protocol and is left up to the hardware manufacturers to implement. This protocol also assumes that the mobile node will always return to its home address, when in fact it might not ever return there. Therefore you have a large amount of rerouting of packets, leading to unnecessary network congestion.

Instead we propose allowing a mobile node to receive a new IP address every time it enters a new subnet. This way, the node is not tied to one particular subnet and avoids having to reroute packets.

A mobile agent, upon entering a new network, registers with the AP, sending over its MIN (Mobile Identification number), which then registers with the server that controls all the nodes and is given a new network address. The AP then makes a delivery Agent for that process which monitors the transactions of the mobile node. This delivery Agent continually keeps track of the status of the mobile node, its location in the network, and handles the local buffering of packets for retransmissions.

Should the mobile node leave the network, the delivery Agent will immediately recognize this, and begin buffering messages for the mobile node. When the mobile node changes its connection point, it again is given a new network address, and a new delivery Agent will be made for him. That delivery agent will then send a change of address to the old Agent telling him the new location. The old delivery Agent will then forward its currently buffered packets and any other incoming packets intended for that node out to the new address, and well as send a change of address packet to the sender of the message informing him where to send any subsequent packets. After a certain time period from when the original change of address packet was received, the old delivery Agent will then terminate.

The general idea is that the most unreliable point of any wireless network is on the immediate network to which the MH is directly connected. For wireless networks, this would be the wireless network between the mobile node and the base station. Therefore, this is the most likely position for errors to occur on the network, so retransmissions are most needed only across this medium, not the whole network. In other words, the message will most likely reach the network's gateway correctly, and then possibly get corrupted or lost while traveling across the medium. Therefore, retransmission is only really needed across the medium, and not the whole network, and this is one of the jobs of the delivery Agent. It buffers up messages and sends them across the medium (provided they aren't corrupted), and then sends an ack back to the sender if it receives one from the mobile node, or retransmits across the medium should an ack not be received. The delivery Agent sits on the AP and resends packets across this medium if the mobile node doesn't receive them. This reduces network traffic requiring multiple retransmissions, as well as quicker delivery in an unstable network. The details of how the WMIP protocol accomplishes this are defined below.

PROTOCOL REQUIREMENTS

A mobile node will no longer keep its static IP address in the dynamic network; it will assume a new one dependent on the new subnet, which it just joined. This requires mobile nodes to register with the gateway(s), which acts as the bridge to the Internet. It also requires that upon registering on the subnet, that a delivery Agent be placed on the AP to monitor the connection and location of the mobile node. This requires a slight change to the registration process and puts slightly more stress on the AP. APs

must now also have some allowances for buffering of messages. But since WMIP guarantees a maximum buffer size, this buffer can be as large or as small as the manufacturers are willing to make it.

There is also a need for the delivery agent to keep track of the location of the MH to know when to commit suicide, so there is a need for control packets to be sent back and forth between the MH and the delivery agent, which will lead to a very slight increase in network traffic over the wireless medium. For cellular networks, no change is needed because the MH already broadcast their position to their BS (base stations) every 2.4 seconds. For 802.11b and HiperLAN2, these control messages would be necessary. Thus, no change is needed to TCP, since it already has all of the capabilities for WMIP built in.

GOALS

The Internet has grown from a small military project to a huge global network of computers in a fantastically small amount of time. One of the drawbacks of this is that there was little time to think of long-term goals of the Internet and long-term scalability of protocols. Routing is an example of this. Routing protocols in IPv4 are lousy and very inefficient, and almost never direct. Therefore, you want as direct routing to a node as possible, without having to reroute more than once when a node changes its point of attachment on the network. WMIP achieves this by informing active session servers of any change in the location of the mobile node to which they are currently sending data to, and has them redirect to the correct location. This decreases network traffic, because the packets will never have to be resent even after the mobile node changes its location in the Internet. This is the first goal of the WMIP protocol.

In wireless networks, the network to which the client is currently attached is usually the most unreliable part of the network, and the main cause of retransmissions. Therefore, most retransmissions are needed from the immediate AP to the MH, not all the way from the source of the message. WMIP attempts to decrease network traffic by having a delivery Agent on the AP, which will handle retransmissions across the immediate network. This is the second goal of the WMIP protocol.

Currently there is no provision for handoffs with today's wireless network protocols. This is all left up to the hardware manufacturers to implement, with no standard in sight. WMIP specifies buffering and the freezing of TCP transmissions during handoffs, which equal a zero packet loss during handoffs, regardless of how long they take and how noisy the network is. The only problem is the slight delay time due to the handoff.

The last and final goal of WMIP protocol is to standardize all the loose ends with wireless network protocols. Wireless networks are slowly taking over, and there are currently very few standards between them. All hardware manufacturers accept packet modulation, so that they are interoperable, but their efficiency changes between manufacturer as determined by how they handle handoffs and inter router scalability. WMIP tries to make an accepted standard by showing its efficiency and how it scales to solve all the loose ends in today's wireless protocols.

ASSUMPTIONS

This protocol makes no new additional constraints from one IP subnet to another. The same IP registration and routing protocols are used with no modifications. This protocol assumes that nodes will generally not change their point of attachment to the Internet more frequently than once per second. This protocol also assumes that APs allow some form of buffering, though it is not actually required. WMIP also assumes that mobile nodes maintain contact with the delivery agent every 2.4 seconds while in use.

NEW ARCHITECTURAL ENTITIES

Mobile Node – A host that changes its point of attachment from one network or Sub-network to another. It treats all IP addresses as temporary and has no 'Home Base' to which it will always return. It is considered completely mobile.

Delivery Agent – A process that runs on an AP or a MDBS which maintains communication with a mobile node and handles delivery of its messages across the medium to which it is attached. The delivery Agent is also responsible for informing other servers of any address changes of the mobile node and redirects any incoming messages to the new location of the mobile node. It is also responsible for freezing the window of the TCP FH sender during times of extreme noisiness as well as during long handoffs.

THE FOLLOWING STEPS PROVIDE A ROUGH OUTLINE OF OPERATION OF WMIP:

- A mobile node enters a new network. It broadcasts a registration message to the gateway attached to the network informing it of its new presence on the network.
- The gateway will then create a delivery agent for the mobile node, which sends a message to the mobile node to establish contact.
- The mobile node will then send a response message back to the delivery Agent containing the previous IP address it was previously known as.

- The delivery Agent will then contact the old delivery Agent and tell it to begin the deregistration procedure.
- The old delivery agent will now forward any packets it receives for the mobile node to the new location, as well as send a change-of-address packet back to the source of the message to inform the source of the new address of the mobile node.
- After the predetermined timeout period has expired, the old delivery agent will remove itself from the AP.
- The new delivery agent, after contacting the old one, begins monitoring the location of the mobile node. Every 2.4 seconds the delivery Agent sends a message to the mobile node to make sure it has not left.
- If the mobile node does not respond after 4 messages, then the delivery agent continues buffering any incoming messages it gets for a predetermined amount of time, or until it starts to use up all of its buffer space. If it does, then it sends a TCP freeze message to the sender. At this point, it assumes that the mobile node has moved, so it waits for either a deregistration message or a timeout, at which point it removes itself.
- If it receives a deregistration message, then it sends its buffered packets to the mobile node's new location, and sends a TCP resume message to the sender, also telling it the new location of the node. It then waits for its timeout.
- If the mobile node has not moved, and a packet comes in for it, then it is intercepted by the delivery Agent. The delivery Agent then sends it across the medium to the mobile node.
- If the mobile node receives it, then it sends an ack back to the delivery Agent. The delivery Agent then sends the ack to the originator of the message, and removes that packet from its buffer.
- If an ack is not received by the delivery Agent, it then resends the message across the medium.
- If an ack is not received after four times, then the delivery Agent considers the mobile node as having moved, and begins buffering any incoming messages for it. The delivery Agent does not send an ack back to the originator of the message, but does continue to buffer the messages coming in.
- If a deregistration message is then sent to the delivery Agent by the mobile node's new delivery Agent, then it begins the deregistration process. Otherwise, its timeout will expire and the delivery Agent will clear its buffer and remove itself from the gateway.

Hence you can see that the delivery Agent does not actually break the TCP connection for client. Instead, it simply handles local retransmissions, as well as the end-to-end synchronization between the FH and it's MH.

As you can see, this is a completely TCP unaware protocol. WMIP takes advantage of current TCP functionality and doesn't introduce anything new to packet types. One thing to note is that WMIP manipulates the RTT for the TCP sender. By not immediately sending the acks until the next one comes in, it keeps the RTT in check by constantly keeping it large. It never allows the RTT to shrink below the actual wireless RTT, so no timeouts should occur, and TCP shouldn't resort to the slow start. WMIP keeps a local timer, which is the estimated RTT for the wireless network, and this is always set to be less than the TCP RTT. We know this is true because we are controlling the TCP RTT. If this is less, than when the local timer expires, the delivery Agent sends a TCP freeze message to the FH until conditions become more favorable, and then the TCP resume message is sent. No loss in window size occurs of the FH, and no timeouts expire on the FH. This keeps us at maximum throughput, with only a slight delay to get over a hump.

AGENT REGISTRATION

When a mobile node starts up after being powered down, it essentially has no idea whether it is on the same network that it was currently on or whether it is on a new network. This presents a major problem because the mobile node MUST be able to tell whether it is on a new network because if it is, it must set up a new delivery Agent on the AP. Also, before a mobile node can send IP datagrams beyond its directly attached subnet, it must discover the address of at least one operational router on that subnet.

This problem can be bypassed by using the ICMP Router Discovery Protocol (Network Working Group). This protocol essentially specifies that each router periodically multicasts a "Router Advertisement" from each of its multicast interfaces, announcing the IP address(es) of that interface. When a mobile node attached to a multicast link starts up, it may multicast a "Router Solicitation" message to query the router for its IP address, rather than waiting for the next periodic ones to arrive. This mechanism allows a mobile node to determine the interface on which it is connected, and also tells the mobile node if he is now attached to a new network. DHCP also allows the mobile node to receive a new IP address on that subnet. ICMP Router Advertisement and ICMP Router

AGENT DEREGISTRATION

An agent will begin its deregistration process when it receives a TCP message sent from the new delivery agent. This message will contain the new address of the delivery agent to be used in the forwarding of messages, and will begin a timer on the delivery Agent for its suicide. The actual payload of the packet is not specified in this paper.

SIMULATION RESULTS

Our simulation was implemented using Network Simulator (McCanne et. al.) and uses many of the assumptions described in (Chan et. al.). Briefly, we assume that the MH moves between the two BS's. The time the MH spends in a cell is exponentially distributed with some mean cell resident time, and the duration of the disconnected period during handoff is called the handoff time (ht). The BS's in turn are connected to a FH via an error-free wired connection. The wireless links are assumed to be error-free. Reno-TCP and Tahoe-TCP in the Network Simulator are used for the simulations.

In Figure 1 below, we see that both Reno and Tahoe are inversely affected by the handoff time. However, the effect of increased handoff time is less dramatic for WIMP (that is, the throughput drops less for WIMP than for either of the other two TCP protocols). The major reason for this is the minimum loss of packets in WIMP, which means most retransmissions are local. Thus, high throughput is maintained. The difference is slightly less dramatic between Tahoe and WIMP probably due to Tahoe's better reaction to multiple losses in a single window.

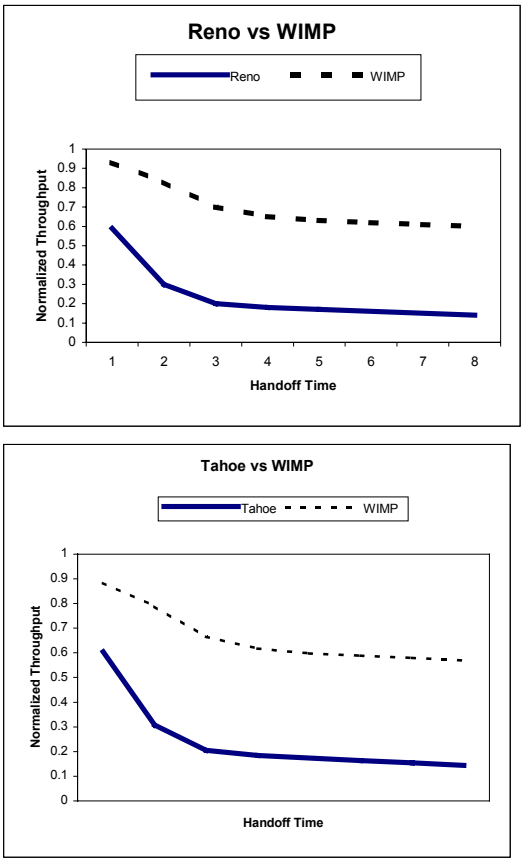

Figure 1 Throughput comparisons for Reno, Tahoe and WIMP.

CONCLUSIONS

The WMIP specification is an interesting protocol because it helps clear up some of the loose ends for wireless networking. Specifically, it helps expand the capabilities of client side routing networking protocols (802.11x, HiperLAN2) to include TCP performance boosts, inter-router handoffs, and intra-router handoffs. Thus, we maintain high throughput without altering the TCP protocol. TCP unaware-ness is maintained, and thus many high load client-side applications can be extended to mobile wireless links without TCP modification.

REFERENCES

Conover, Joel, "Anatomy of IEEE 802.11b Wireless", http://www.networkcomputing.com/1115/1115ws2.html, August 7, 2000

The Internet Weather Report: Animated maps of current Internet lag, http://www2.aus.us.mids.org/weather/

Network Working Group Request for Comments: 1256, ICMP Router Discovery Messages, http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc1256.html

Gerla, M., Tang, K. and Bagrodia, R. , "TCP Performance in Wireless Multi-hop Networks", Proceedings of IEEE WMSCA'99, New Orleans.

Balakrishnan, H., Padmanabhan, V. , Seshan, S. , Katz, R. , " A Comparison of Mechanisms for Improving TCP Performance over Wireless Links". In Proc. ACMSIGCOMM '96, pp. 256-269, 1996.

Fitzek, F., Mota, E., Ewers, E. and Wolisz, A. , "An efficient approach for speeding up simulation of wireless networks", WMC2000

McCanne, S., Floyd, S., and K. Fall, "NS -LBNL Network Simulator".

Chan, A., Tsang, D. and Gupta, S., "Impacts of Handoff on TCP Performance in Mobile Wireless Computing", Proceedings of ICPWC97

Manzoni, P. , Ghosal , D. and Serazzi, G. "A Simulation Study of the Impact of Mobility on TCP/IP", IEEE JSAC, Vol13, No. 5, P858-6, June 1995

Chan, A., Tsang, D. and Gupta, S., "TCP Transmission Control Protocol over Wireless Links ", Proceedings of VTC'97, Phoenix, USA, May 1997

BIOGRAPHY

Robert Signorile is currently an Associate Professor and Chair in the Computer Science Department of Boston College. His research interests include networks, distributed systems, and multi-model simulation. He has published regularly in applied simulation and simulation methodology.

Kevin Lester is a graduate of the Computer Science Department of Boston College.